

阿里云ApsaraDB诊断报告

1 文档简介

1.1 文档目的

本文档描述了阿里云客户支持服务对当前问题状态的理解和认识，并给出相关建议和结论。

本文档既用于让客户对当前问题有全面的了解，也用于协调各相关人员对问题状态达成统一的认识。

1.2 问题类别定义

编号	类别	说明
1.	应用问题	应用问题包含开发、设计等问题。
2.	操作使用问题	操作使用问题包含操作不规范,操作失误等问题。
3.	产品问题	产品问题包含软件产品、硬件产品及第三方产品问题。
4.	未知问题	未知问题指当前无法判断的问题类别。

1.3 问题严重级别定义

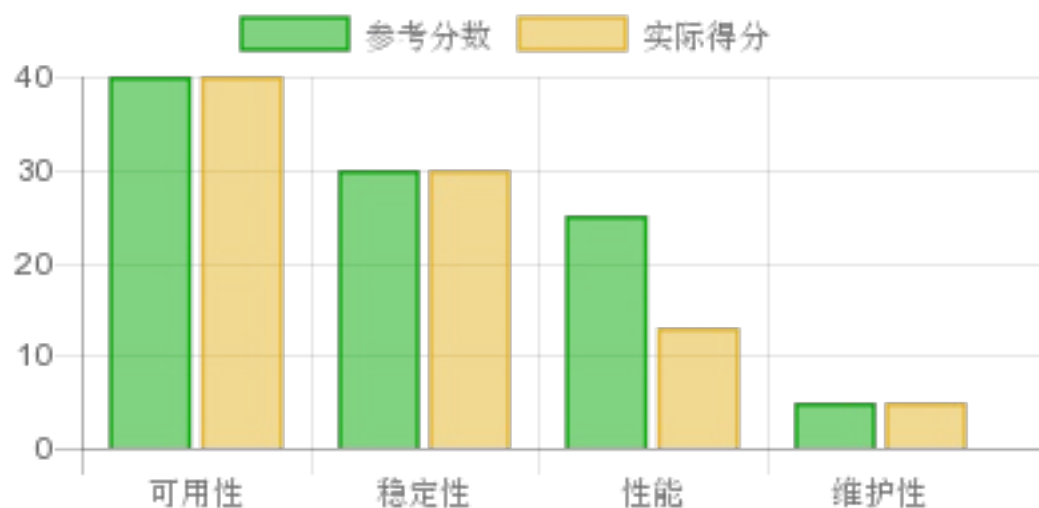
以下为我们对系统环境中问题的严重级别定义：

编号	类别	说明
S1	系统失败，任何处理都不能继续。	
S2	不能够继续选定的任务处理。	
S3	受限制的功能能力，但处理可以继续。	
S4	目前可以忽略的小问题。	

2 规格配置

配置项	规格名称
实例名	rr-2ze83o0z0lc4tq346
数据库类型	mysql
数据库版本	5.6
规格名称	MySQL新规格4核8G
存储空间(M)	51200
最大连接数	2000
最大IOPS	5000
QPS参考值	999999

综合评分：88分



3 系统状态

时间范围 2017-06-21 08:35 至 2017-06-21 15:35



资源名称	资源状态	最大值	最小值	平均值	参考值
CPU利用率	良好	37.40%	3.50%	7.52%	20%~40%
内存利用率	空闲	35.60%	34.80%	35.22%	40%~80%
磁盘IOPS	空闲	1.10%	0.10%	0.31%	10%~40%
磁盘空间	空闲	12.10%	11.80%	11.94%	30%~60%
连接数	空闲	2.30%	0.90%	1.64%	10%~40%

4 问题与建议

4.1 死锁

没有发现死锁。

4.2 锁等待

没有发现锁等待。

4.3 实例安全

没有发现相关问题。

4.4 连接数

没有发现相关问题。

4.5 内存

没有发现相关问题。

4.6 实例空间

没有发现相关问题。

5 SQL分析

5.1 慢SQL分析

5.1.1 SQL-1

SQL语句	<pre>SELECT COUNT(*) FROM (SELECT pu.upload_image_id, pu.case_info_id, pu.loss_item_id, pu.case_no, pu.upload_time , pu.user_code, pu.mobile, pu.oss_key, pu.license_no, pu.shot_time , pu.shot_latitude, pu.shot_longitude, pu.img_name, pu.img_path, pu.img_type , pu.lock_flag, pu.break_point_flag, pu.stay_upload, pu.remark, pu.create_by , pu.create_time, pu.modify_by, pu.modify_time, l.loss_name AS 'prpmLossItem.lossName', l.loss_type AS 'prpmLossItem.lossType' , l.loss_id AS 'prpmLossItem.lossId' FROM prpm_upload_image pu JOIN prpm_item_img_rela pi ON pu.upload_image_id = pi.upload_image_id JOIN prpm_loss_item l ON pi.loss_item_id = l.loss_item_id WHERE 1 = 1 AND l.org_code = ? AND pu.upload_time >= ? AND pu.upload_time <= ? ORDER BY pu.upload_time DESC) tmp_count</pre>
建议	<ol style="list-style-type: none">1. 行扫描 行扫描与行返回之比为1685813。2. order by 该语句使用了"order by"。注意检查是否有效利用到了索引。3. 聚合函数 该语句使用了聚合函数。执行次数并不高，但单次执行时间超过5秒。需要注意该大查询对系统整体性能的影响。

5.1.2 SQL-2

SQL语句	<pre>SELECT pu.upload_image_id, pu.case_info_id, pu.loss_item_id, pu.case_no, pu.upload_time , pu.user_code, pu.mobile, pu.oss_key, pu.license_no, pu.shot_time , pu.shot_latitude, pu.shot_longitude, pu.img_name, pu.img_path, pu.img_type , pu.lock_flag, pu.break_point_flag, pu.stay_upload, pu.remark, pu.create_by , pu.create_time, pu.modify_by, pu.modify_time, l.loss_name AS 'prpmLossItem.lossName', l.loss_type AS 'prpmLossItem.lossType' , l.loss_id AS 'prpmLossItem.lossId' FROM prpm_upload_image pu JOIN prpm_item_img_rela pi ON pu.upload_image_id = pi.upload_image_id JOIN prpm_loss_item l ON pi.loss_item_id = l.loss_item_id WHERE 1 = 1 AND l.org_code = ? AND pu.upload_time >= ? AND pu.upload_time <= ? ORDER BY pu.upload_time DESC LIMIT ?, ?</pre>
建议	<ol style="list-style-type: none">1. 行扫描 行扫描与行返回之比为165544。2. order by 该语句使用了"order by"。注意检查是否有效利用到了索引。3. 索引 该语句没有使用聚合函数和模糊查询，行扫描与行返回之比很高。创建合理索引会带来很大的优化空间。

5.1.3 SQL-3

SQL语句	<pre>SELECT pu.upload_image_id, pu.case_info_id, pu.loss_item_id, pu.case_no, pu.upload_time , pu.user_code, pu.mobile, pu.oss_key, pu.license_no, pu.shot_time , pu.shot_latitude, pu.shot_longitude, pu.img_name, pu.img_path, pu.img_type , pu.lock_flag, pu.break_point_flag, pu.stay_upload, pu.remark, pu.create_by , pu.create_time, pu.modify_by, pu.modify_time, l.loss_name AS 'prpmLossItem.lossName', l.loss_type AS 'prpmLossItem.lossType' , l.loss_id AS 'prpmLossItem.lossId' FROM (SELECT pp.*, pi.loss_item_id AS itemId FROM prpm_upload_image pp INNER JOIN prpm_item_img_rela pi ON pp.upload_image_id = pi.upload_image_id WHERE 1 = 1 AND pp.case_no LIKE CONCAT(CONCAT(?, ?), ?) AND pp.upload_time >= ? AND pp.upload_time <= ? ORDER BY pp.upload_time DESC) pu INNER JOIN prpm_loss_item l WHERE pu.itemId = l.loss_item_id AND l.org_code = ?</pre>
建议	<ol style="list-style-type: none"> 1. 行扫描 行扫描与行返回之比为123976。 2. order by 该语句使用了"order by"。注意检查是否有效利用到了索引。 3. 索引 该语句没有使用聚合函数和模糊查询，行扫描与行返回之比很高。创建合理索引会带来很大的优化空间。

5.1.4 SQL-4

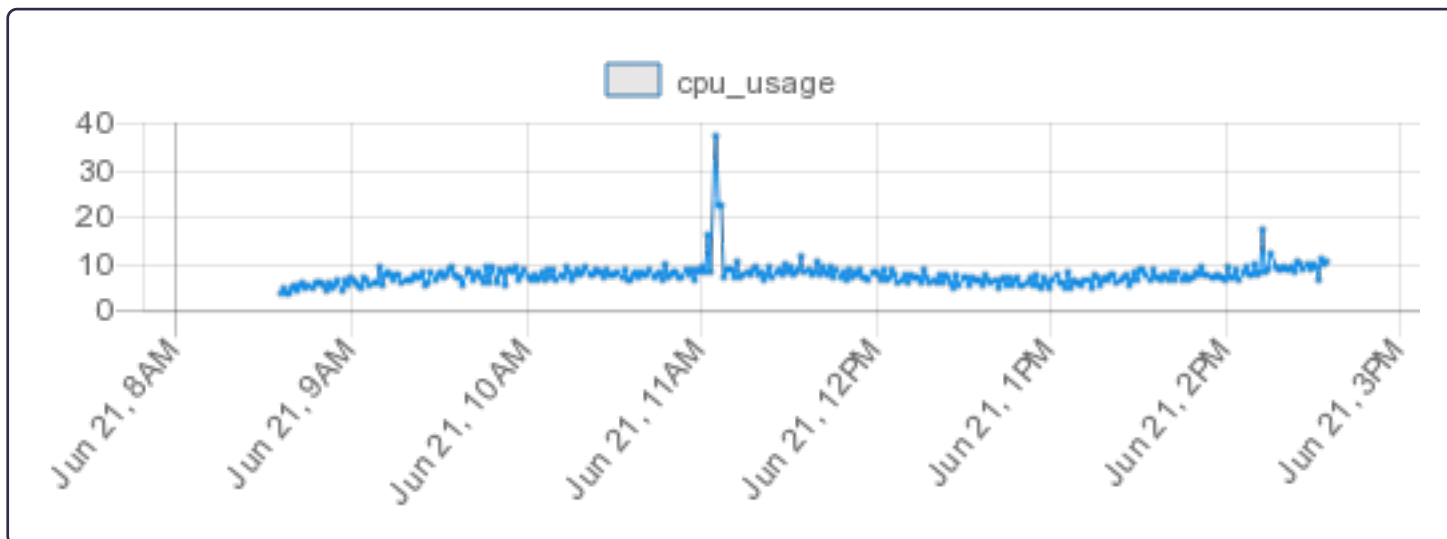
SQL语句	<pre>SELECT COUNT(*) FROM (SELECT pu.upload_image_id, pu.case_info_id, pu.loss_item_id, pu.case_no, pu.upload_time , pu.user_code, pu.mobile, pu.oss_key, pu.license_no, pu.shot_time , pu.shot_latitude, pu.shot_longitude, pu.img_name, pu.img_path, pu.img_type , pu.lock_flag, pu.break_point_flag, pu.stay_upload, pu.remark, pu.create_by , pu.create_time, pu.modify_by, pu.modify_time, l.loss_name AS 'prpmLossItem.lossName', l.loss_type AS 'prpmLossItem.lossType' , l.loss_id AS 'prpmLossItem.lossId' FROM prpm_upload_image pu JOIN prpm_item_img_rela pi ON pu.upload_image_id = pi.upload_image_id JOIN prpm_loss_item l ON pi.loss_item_id = l.loss_item_id WHERE 1 = 1 AND l.org_code = ? AND pu.case_no LIKE CONCAT(CONCAT(?, ?), ?) AND pu.upload_time >= ? AND pu.upload_time <= ? ORDER BY pu.upload_time DESC) tmp_count</pre>
建议	<ol style="list-style-type: none"> 1. 行扫描 行扫描与行返回之比为1654219。 2. order by 该语句使用了"order by"。注意检查是否有效利用到了索引。 3. 聚合函数 该语句使用了聚合函数。

5.1.5 SQL-5

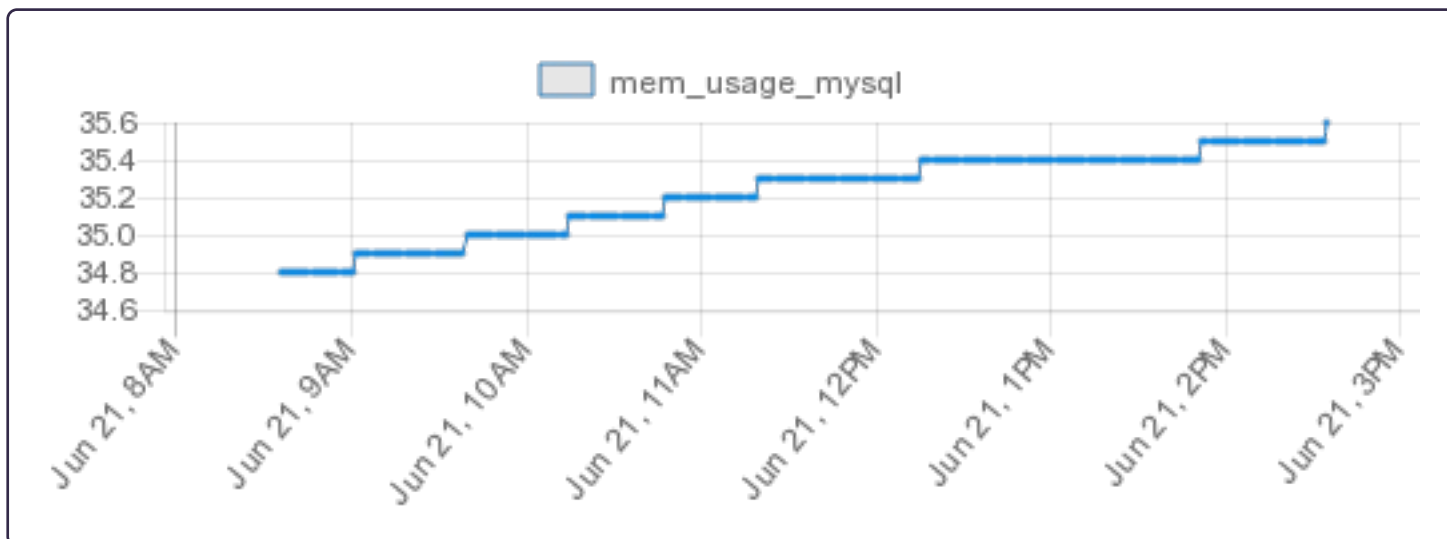
SQL语句	<pre>SELECT pu.upload_image_id, pu.case_info_id, pu.loss_item_id, pu.case_no, pu.upload_time , pu.user_code, pu.mobile, pu.oss_key, pu.license_no, pu.shot_time , pu.shot_latitude, pu.shot_longitude, pu.img_name, pu.img_path, pu.img_type , pu.lock_flag, pu.break_point_flag, pu.stay_upload, pu.remark, pu.create_by , pu.create_time, pu.modify_by, pu.modify_time, l.loss_name AS 'prpmLossItem.lossName', l.loss_type AS 'prpmLossItem.lossType' , l.loss_id AS 'prpmLossItem.lossId' FROM prpm_upload_image pu JOIN prpm_item_img_rela pi ON pu.upload_image_id = pi.upload_image_id JOIN prpm_loss_item l ON pi.loss_item_id = l.loss_item_id WHERE 1 = 1 AND l.org_code = ? AND pu.case_no LIKE CONCAT(CONCAT(?, ?), ?) AND pu.upload_time >= ? AND pu.upload_time <= ? ORDER BY pu.upload_time DESC LIMIT ?, ?</pre>
建议	<ol style="list-style-type: none">1. 行扫描 行扫描与行返回之比为165420。2. order by 该语句使用了"order by"。注意检查是否有效利用到了索引。3. 索引 该语句没有使用聚合函数和模糊查询，行扫描与行返回之比很高。创建合理索引会带来很大的优化空间。

6 性能曲线

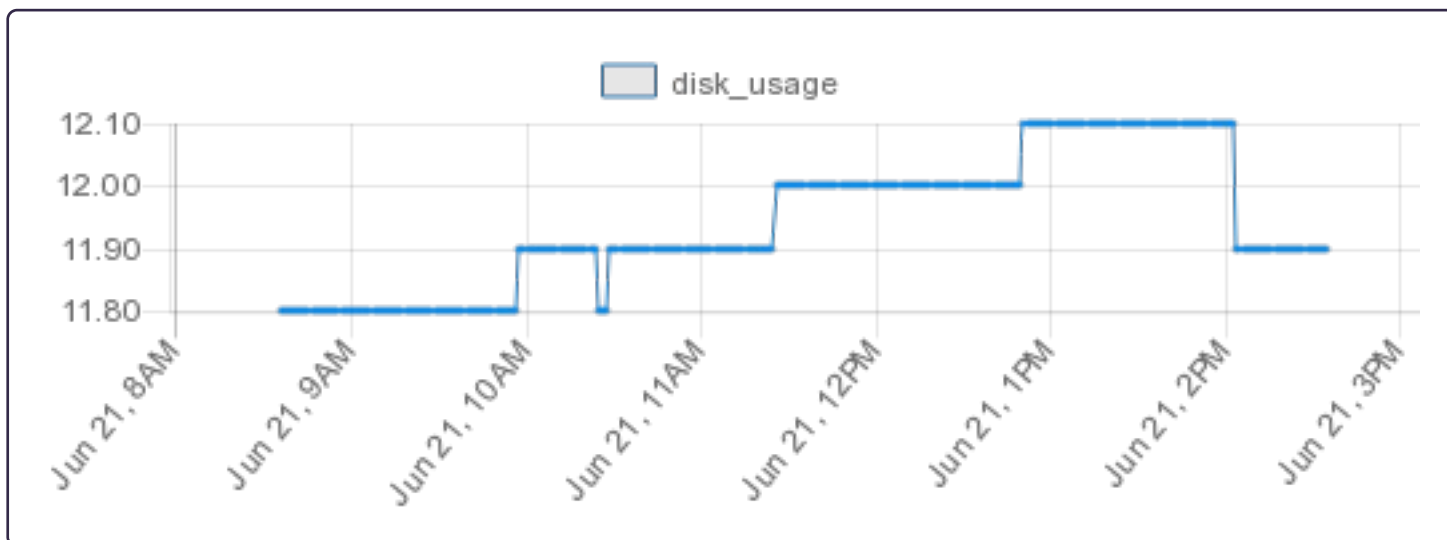
6.1 CPU利用率



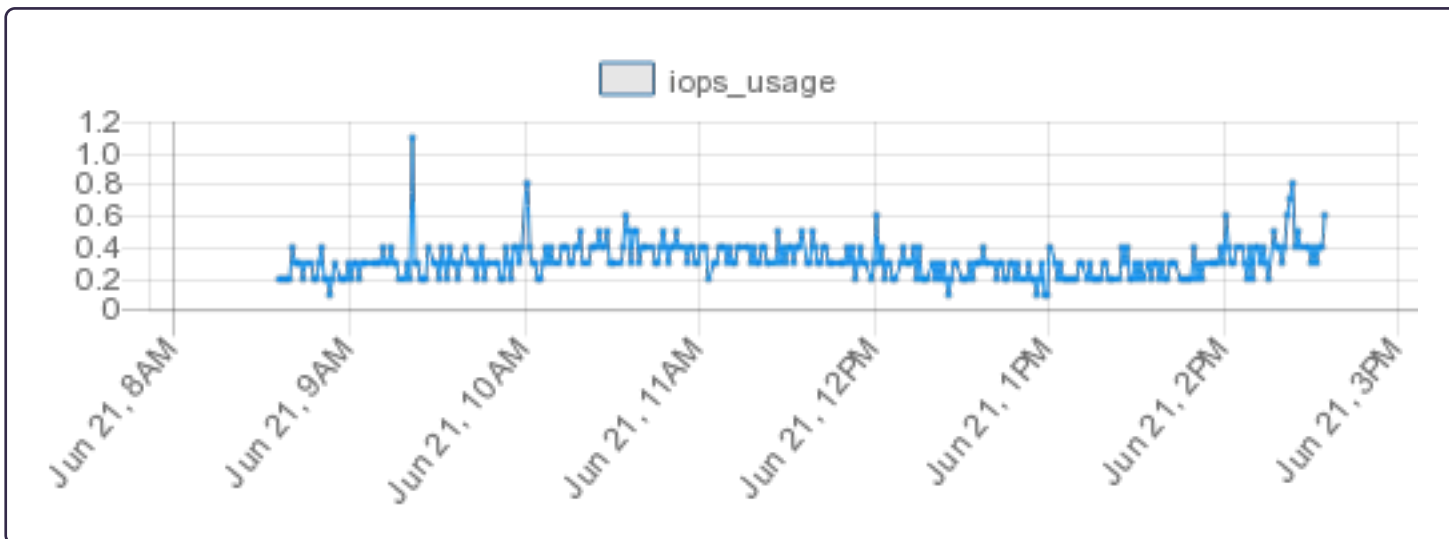
6.2 内存利用率



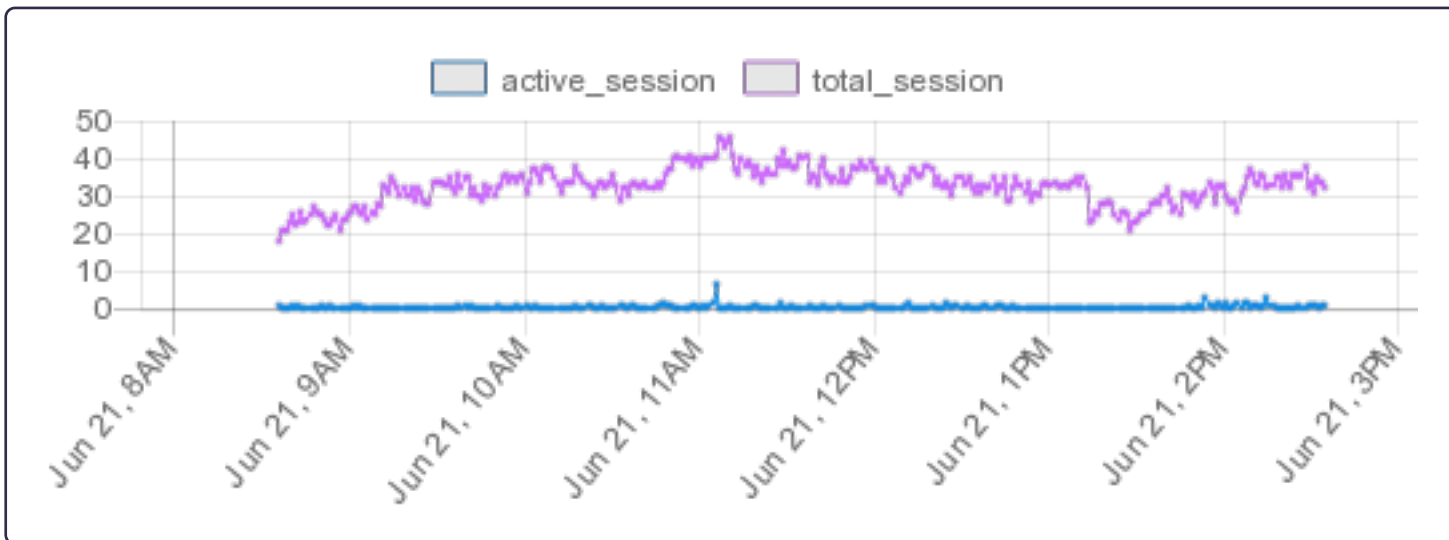
6.3 磁盘空间



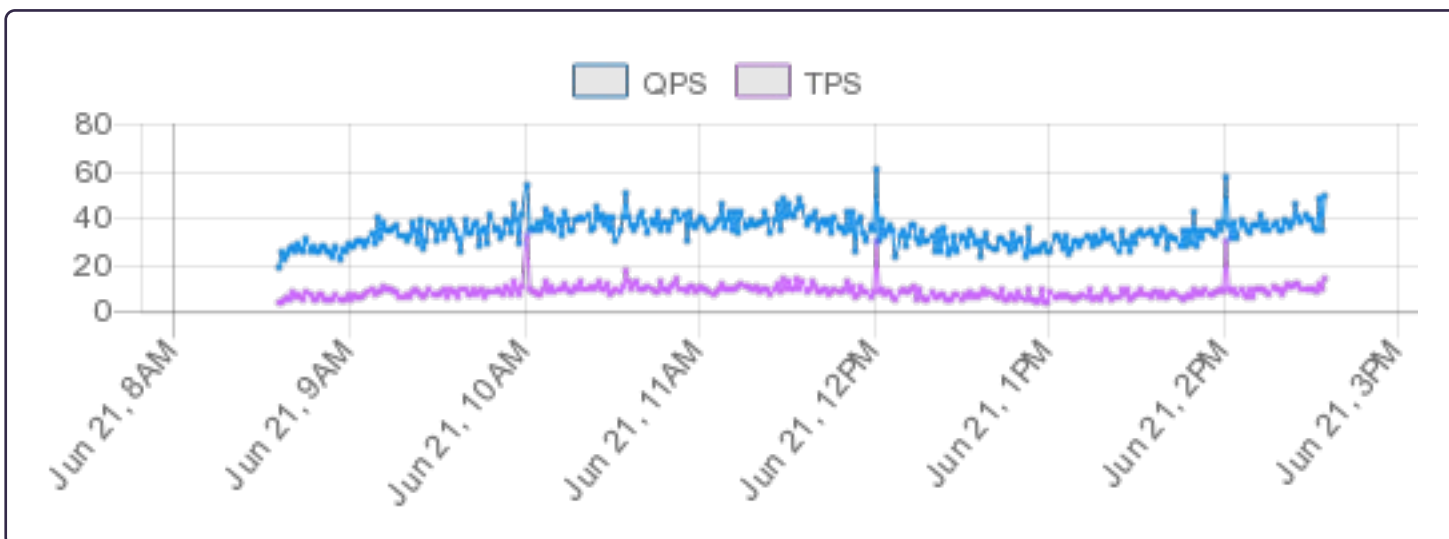
6.4 IOPS利用率



6.5 会话



6.6 QPSTPS



6.7 InnoDB

